ECE/CS 5565 Project 5

Name: Daksh Dave

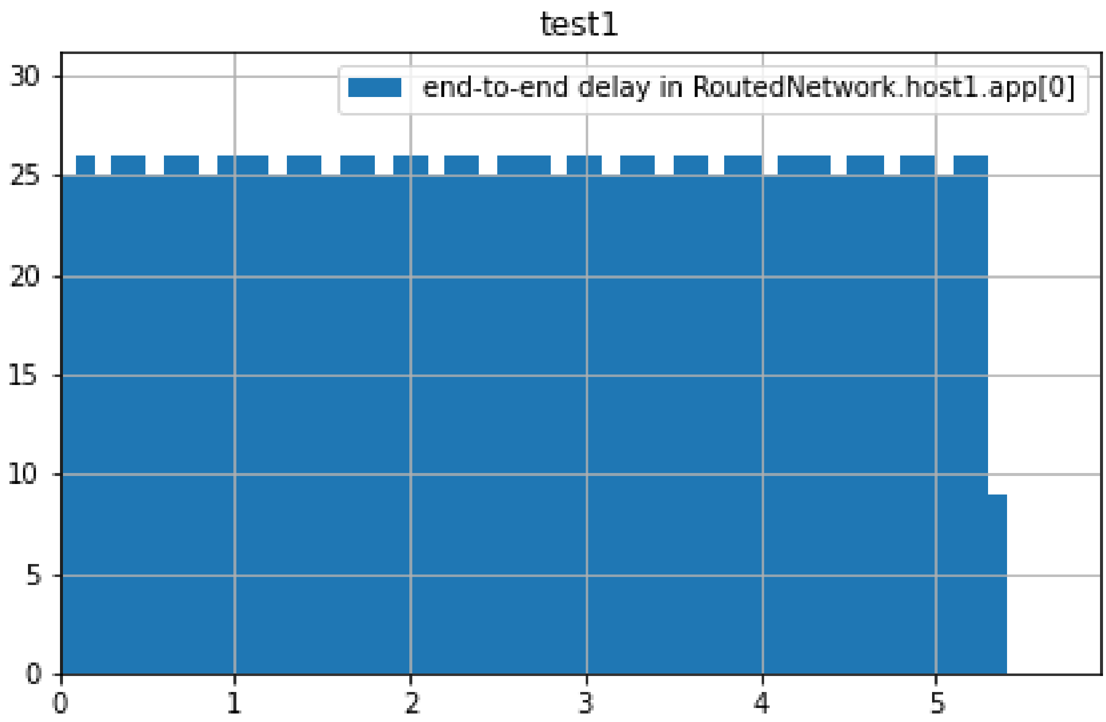
Email: ddave@vt.edu

1.1

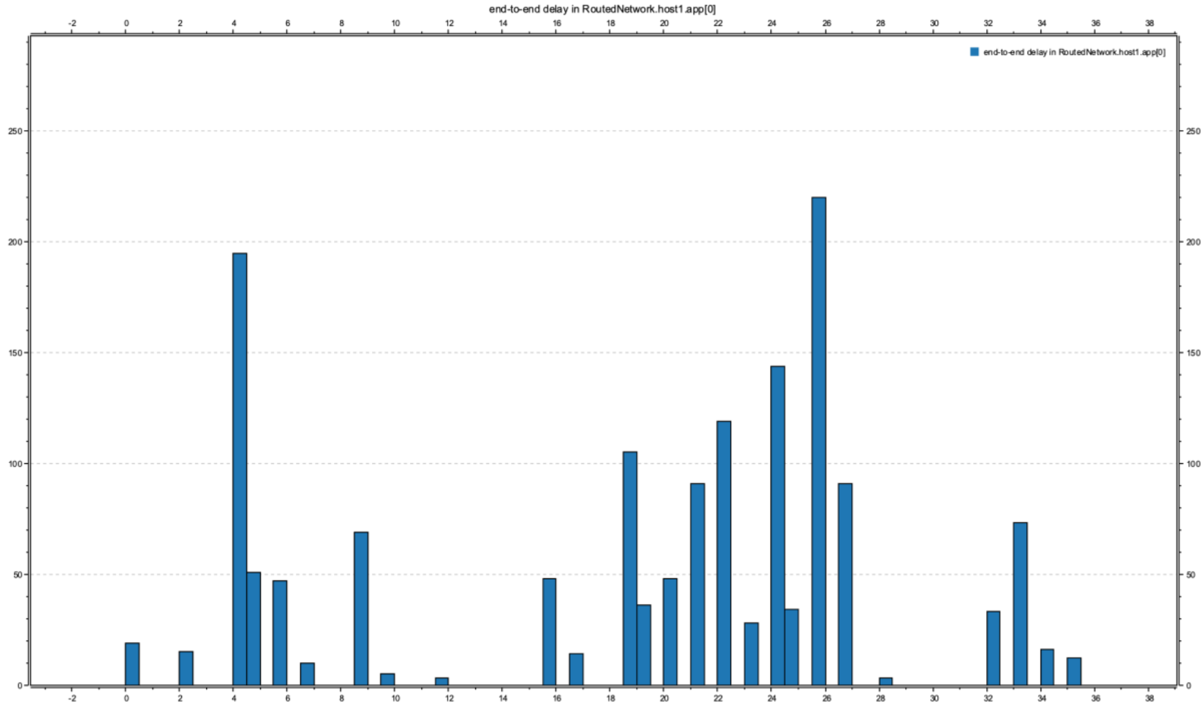
| Run | TCP | R1-R3 PER | Mean Delay(s) | StdDevDelay (s) | Throughput(kbps) |
|---|---|---|---|---|---|
| **1** | NoCongestionControl | 0.00 | 2.669 | 1.540 | 3250kbps |
| **2** | NoCongestionControl | 0.01 | 18.91 | 9.264 | 499kbps |
| **3** | Tahoe | 0.00 | 2.668 | 1.540 | 3256kbps |
| **4** | Tahoe | 0.01 | 3.889 | 2.168 | 2115kbps |

1.2

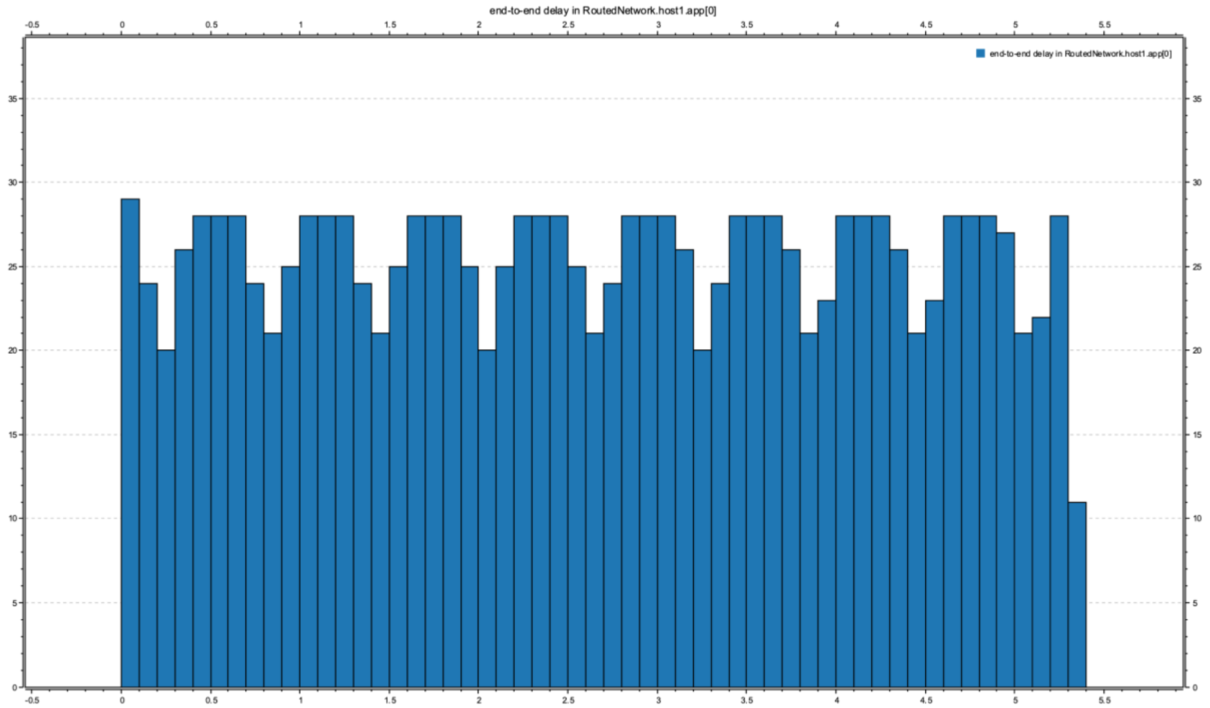
Run1- NoCongestionControl-0.00



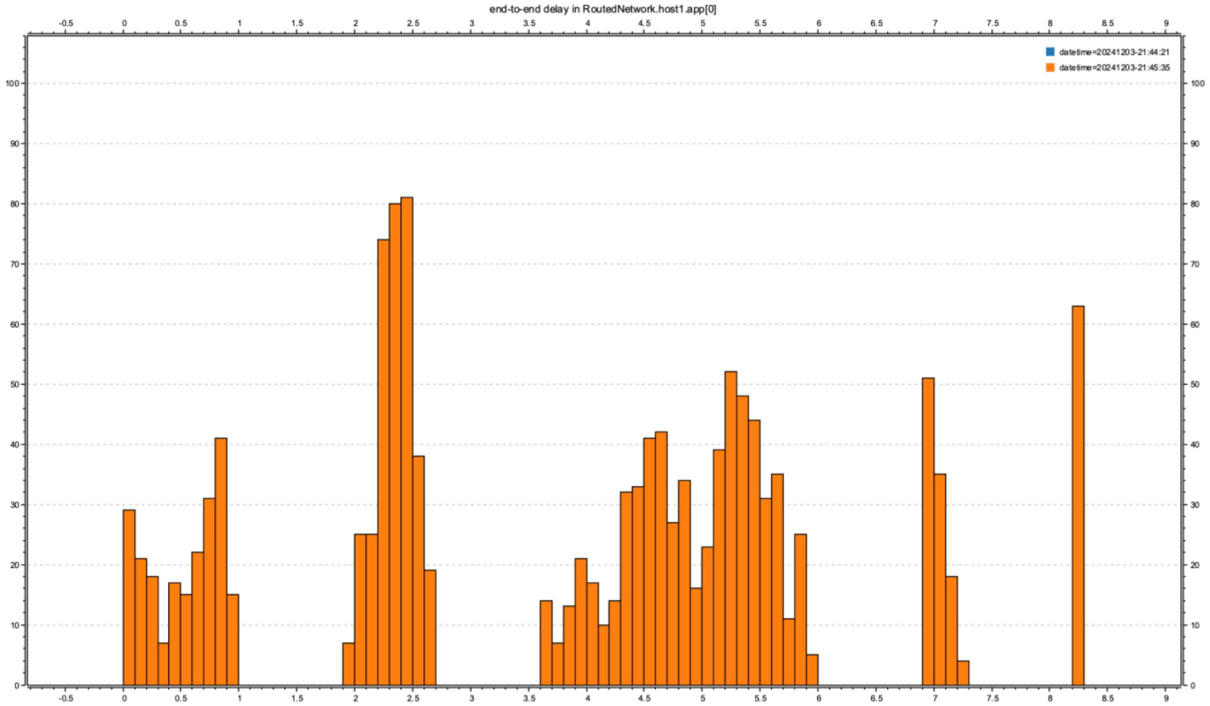

Run2- NoCongestionControl-0.01




Run 3-Tahoe-0.00




Run4-Tahoe-0.01




1.3(a)

Observation: Both Run 1 (No Congestion Control, PER = 0.0) and Run 3 (Tahoe, PER = 0.0) have nearly identical mean delay (2.669s vs. 2.668s) and standard deviation (1.540s).

Neither performs markedly better because, in the absence of packet errors, the lack of congestion does not significantly affect delay.

1.3(b)

Observation: Run 2 (No Congestion Control, PER = 0.01) has a much higher mean delay (18.91s) and standard deviation (9.264s) compared to Run 4 (Tahoe, PER = 0.01, mean delay = 3.889s, std dev = 2.168s).

Explanation: TCP Tahoe introduces congestion control mechanisms, including slow start and retransmissions, which mitigate packet loss effects. In contrast, "No Congestion Control" results in severe delays due to unregulated retransmissions.

Run 4 performs markedly better due to Tahoe's congestion control.

1.3(c)

Observation: Run 2 (PER = 0.01) has a significantly higher delay (18.91s vs. 2.669s) compared to Run 1 (PER = 0.0).

Explanation: The introduction of packet errors in Run 2 results in retransmissions, causing congestion and increased delays.

Run 1 performs better as there are no errors, while Run 2 suffers due to the absence of congestion control mechanisms.

1.3(d)

Observation: Run 4 (PER = 0.01) has a higher delay (3.889s) than Run 3 (PER = 0.0, 2.668s).

Explanation: Packet errors in Run 4 cause retransmissions, but Tahoe mitigates this impact through its congestion control features (e.g., congestion avoidance). Run 4's performance is significantly better than Run 2 because Tahoe effectively handles packet loss.

Run 3 performs better because there are no errors, but Run 4 demonstrates robust handling of errors due to Tahoe.

1.3(e)

Packet errors drastically increase delay when no congestion control is employed.

Tahoe's mechanisms significantly reduce the adverse effects of packet loss.

As the environment becomes error-prone, congestion control mechanisms like those in TCP Tahoe become critical in maintaining acceptable performance. These findings highlight the importance of incorporating congestion control.

2. 1

a) Frame No.-59

b) Window size = 20440

Bytes in flight=20440

Max window size =20440 = Advertised Window size

From wireshark it can be ascertained that

(Last Byte Sent – Last Size Acked) = Bytes in flight = 20440

Therefore, effective window size=20440-20440=0

c)

Bytes in flight=18980

Max Window Size= 20440

Effective Window Size = 20440 – 18980 = 1460

d) TCP flow control ensures that a client does not send more data than the receiver can handle at any given time. When the bytes in flight (data sent but not yet acknowledged) reach the Maximum Window Size, the sender pauses further transmissions. This mechanism prevents the client from overwhelming the receiver by respecting the constraints imposed by the receiver's available buffer size. Once the receiver processes and acknowledges more data, the sender can resume sending additional data.

2.2

a) Frame No. -233

(b) In Frame 232, Host 1 acknowledges receipt of all data up to byte 73000 by sending an acknowledgement number of 73001. In the subsequent Frame 233, the server sends a data segment starting with the relative sequence number 74461, covering bytes 74461 to 75920. However, Wireshark reports a warning: "TCP Previous segment not captured." This indicates that the data between 73001 and 74460 was not received by Host 1, suggesting a gap or packet loss.

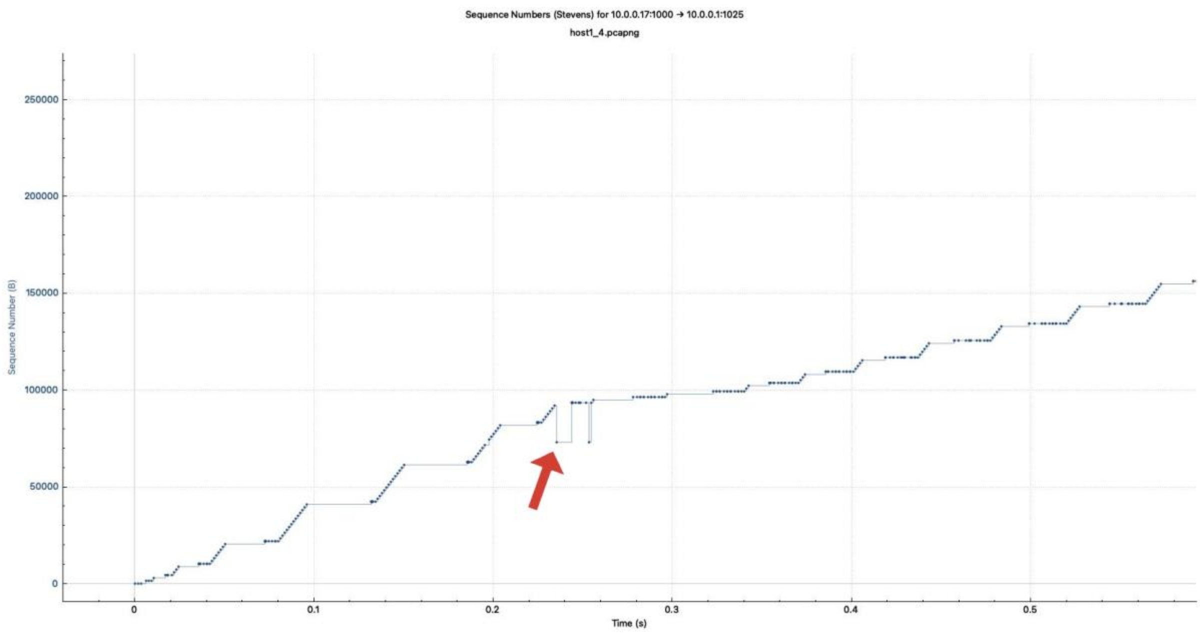
(c) Wireshark labels the segment as "TCP Dup ACK 232#1," signifying that this is a duplicate acknowledgement for sequence number 73001, referencing Frame 232. This duplicate ACK signals that Host 1 has successfully received data up to byte 73000 but hasn't received the following byte (73001) due to packet loss. As a result, Host 1 retransmits the same acknowledgement to prompt the retransmission of the missing data.

(d) After the 6th duplicate acknowledgement, in Frame 287, the server retransmits the missing segment to address the packet loss.

(e) Wireshark indicates that the server at 10.0.0.1 performs a fast retransmission for the segment starting at sequence number 73001. This fast retransmission is triggered by the detection of multiple duplicate ACKs. TCP uses these multiple ACKs to differentiate between actual packet loss and delays, ensuring efficient and timely recovery of lost data.

2.3

a)




(b)

The operation mode observed is the Slow Start phase, which is a key aspect of TCP Tahoe behaviour. This phase is characterized by a steady, exponential growth of the sequence numbers until packet loss occurs. The linear increase seen in the sequence numbers indicates that the connection is in Slow Start, aiming to avoid network congestion while efficiently ramping up data transmission.

(c)

TCP operates in the Congestion Avoidance phase, punctuated by retransmission events caused by packet loss. This behaviour matches TCP Tahoe's congestion control mechanism, where the congestion window is reset, and the protocol falls back to Slow Start after detecting packet loss. The graph's sharp increases and sudden drops reflect this pattern of congestion control, showing the transitions between the recovery and congestion avoidance phases.

**3. Summary**

No problems were encountered in the project.